Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method in a hardware environment for ~~simulating~~ validating a design for a system which comprises a software element, and first and second hardware components, the software element being for execution on the second hardware component, and the first and second hardware components being operable to interact with one another, the method comprising the steps of:

simulating operation of the first hardware component in a first simulation in a hardware environment; [[and]]

simulating the software element and the second hardware component in a second simulation using a software model embedded within the hardware environment; and

analyzing the first and second simulations to validate the design for the system,

wherein the first simulation and the second simulation are implemented in separate processing threads within the hardware environment, and

wherein the first and second simulation run asynchronously [[,]] with the second simulation running ahead of the first simulation, allowing the software model to control the first simulation of the first hardware component and allowing for more rapid simulation of software instructions in the software model.

2. (Canceled)

3. (Previously Presented) A method as claimed in claim 1, wherein a number of clock cycles of the first simulation and the second simulation is synchronised with a reference clock.

4. (Canceled)

5. (Original) A method as claimed in claim 1, further comprising:

performing operations in the first simulation to set up an inter-process communications protocol connection therein;

connecting the second simulation to the inter-process communications protocol connection in the first simulation;

connecting a software debugger to the second simulation; and

controlling the first simulation from the software debugger via the second simulation using the inter-process communications protocol.

6. (Original) A method as claimed in claim 1, further comprising:

performing operations in the first simulation to set up an inter-process communications protocol connection therein;

connecting a software debugger to the communications protocol connection; and

controlling the first simulation from the software debugger using the inter-process communications protocol.

7. (Original) A method as claimed in claim 5 or 6, wherein the inter-process communications protocol is TCP/IP and the connection is a TCP/IP socket.

8. (Original) A method as claimed claim 1, wherein the second hardware component includes a processor.

9. (Original) A method as claimed in claim 8, wherein the processor is an embedded processor.

10. (Original) A method as claimed in claim 1, wherein the hardware component includes processor peripheral devices.

11. (Original) A method as claimed in claim 10, wherein the peripheral devices are embedded.

12. (Original) A method as claimed in claim 1, wherein the first simulation is implemented using a hardware description language (HDL) simulation environment.

13. (Original) A method as claimed in claim 1, wherein the second simulation is implemented using a C model.

14. (Original) A method as claimed in claim 1, wherein the first hardware component is a programmable logic device.

15. (Currently Amended) A method in a hardware environment for controlling a simulation of a system using a software debugger, the simulation useful for validating a design of the system, wherein the system comprises a software element, and first and second hardware components, [[;]] the software element being for execution on the second hardware component and the first and second hardware components being operable to interact with one another, the method comprising the steps of:

simulating the first hardware component in a first simulation in the hardware environment;

simulating the software element and the second hardware component in a second simulation using a software model embedded within the hardware environment;

performing operations to set up an inter-process communications protocol connection;

connecting the software debugger to the software model of the second simulation embedded in the hardware environment; and

controlling the first simulation of the first hardware component from the software debugger through the software model of the second simulation using the inter-process communications protocol. ~~, the software debugger having no knowledge of the connection between the first and second simulations.~~

16. (Original) A method as claimed in claim 15, further comprising the step of:

connecting the software debugger to inter-process communications protocol connection.

17. (Canceled)

18. (Original) A method as claimed in claim 15, wherein the inter-process communications protocol is TCP/IP and the connection is a TCP/IP socket.

19. (Original) A method as claimed in claim 15, wherein the step of simulating the second hardware component comprises simulating a processor and one or more peripheral devices with which the one or more processors interact directly.

20. (Original) A method as claimed in claim 15, wherein the first simulation and the second simulation are implemented in separate processing threads.

21. (Original) A method as claimed in claim 15, wherein the first simulation and the second simulation run asynchronously.

22. (Original) A method as claimed in claim 15, wherein the first simulation and the second simulation are synchronised with a reference clock.

23. (Original) A method as claimed in claim 15 wherein the first and second simulations are implemented in respective different simulation environments.

24. (Original) A method as claimed in claim 15, wherein the second hardware component includes embedded processors.

25. (Original) A method as claimed in claim 15, wherein the second hardware component includes embedded peripheral devices.

26. (Original) A method as claimed in claim 15, wherein the first simulation is implemented using a hardware description language (HDL) simulation environment.

27. (Original) A method as claimed in claim 15, wherein the second simulation is implemented using a C model.

28. (Original) A method as claimed in claim 15, wherein the first hardware component is a programmable logic device.

29. (Currently Amended) A method for providing an I/O interface for a simulation model to allow the simulation of interactive programs in a hardware environment for use in system validation, the method comprising:

simulating a software element using a software model in a first processing thread in the hardware environment;

simulating an embedded input/output device within the simulation model to produce an input/output device model in a second processing thread;

connecting the input/output device model to a terminal emulator using an inter-process communications protocol;

running an interactive program in the terminal emulator to transfer information to the input/output device model; and

polling the input/output device model for the transferred information using the software model.

30. (Original) A method as claimed in claim 29, the method further comprising:

providing separate processing threads for the embedded input/output device to allow concurrent user inputs and outputs.

31. (Original) A method as claimed in claim 29, wherein the inter-process communications protocol is TCP/IP.

32. (Original) A method as claimed in claim 29, wherein the input/output device is a UART device.

33. (Original) A method as claimed in claim 29, wherein the input/output device is an Ethernet MAC device.